

Wright State University

CORE Scholar

---

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

---

Fall 2006

## CEG 221: Advanced C Programming for Engineers

Robert Helt

*Wright State University - Main Campus*

Follow this and additional works at: [https://corescholar.libraries.wright.edu/cecs\\_syllabi](https://corescholar.libraries.wright.edu/cecs_syllabi)



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

---

### Repository Citation

Helt, R. (2006). CEG 221: Advanced C Programming for Engineers. .  
[https://corescholar.libraries.wright.edu/cecs\\_syllabi/49](https://corescholar.libraries.wright.edu/cecs_syllabi/49)

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact [library-corescholar@wright.edu](mailto:library-corescholar@wright.edu).

# Syllabus

## CEG 221 Advanced C Programming for Engineers

### Section 1 – Fall 2006

T - Th 6:05 – 7:45 p.m. in Russ Engineering Center Room RC152A

**Description:** This course introduces advanced constructs, algorithms, and data structures in the C programming language. Emphasis is on problem solving and techniques useful to engineers. Topics include functions, array, pointers, structures as well as sorting algorithms, linked lists, complex numbers, stacks, queues, hash tables, and binary trees. 4 credit hours. Prerequisite: CEG220 (Introduction to C Programming for Engineers).

**Instructor:** Robert Helt, Russ Engineering Center Room RC160 (Student Lounge Area) E-mail: rhelt@wright.edu Hours: 4:30 to 5:30 p.m. M – W or immediately following class. Other hours arranged by appointment through email requests.

**Textbooks:**

C: The Complete Reference, Fourth Edition, Herbert Schildt, Osborne/McGraw-Hill, 2000.

**Software:** The Dev-Cpp 4.9 C Compiler is available on the lab computers as well as a CD from the instructor. Alternate C compilers are Visual Studio.net, Visual Studio 6.0, and the UNIX GNU C compiler. Other C compilers must be approved by the instructor.

**Grading:**

**Programming Projects:** Programming projects are assigned on Monday. Each project is due the following Monday, one week from when it was assigned. Each programming assignment is worth 10 points. Programming assignments will be graded as *Satisfactory* or *Unsatisfactory*! For a project to be satisfactory, 1) the source code file must contain the required header information, 2) the source code must meet style and documentation guidelines, 3) the program must compile and run without warnings or errors and produce the correct output, and 4) the project report must adequately address all the required areas. If a project is graded as satisfactory, six to ten points will be awarded, depending on how well the project meets the specifications and grading criteria. Each time a programming project is turned in and graded as unsatisfactory, one point will be lost. Unsatisfactory projects must be resubmitted not later than **midnight** two days after it is returned. If any portion of the assignment is turned in late, one-half point will be deducted for each day it is late. Any unsatisfactory assignment that is finally graded satisfactory will not be worth less than 6 points. The programming projects will comprise 25% of the grade. **All six projects must be turned in by the last day of classes for the quarter and completed with a grade of *Satisfactory* to pass the course!**

**Examinations:** There will be two **One-Hour Exams** that will comprise 40% of the final grade. There will be one **Comprehensive Final Exam** worth 35% of the final grade. All exams will be closed book, closed notes, no electronic devices in view.

**Grades:** A: 100-90, B: 89-80, C: 79-70, D: 69-60, F: 59-0 or all programming projects not completed with a grade of *Satisfactory*.

**Policy:** All programming projects will be turned in electronically by email and are due before **midnight** on the dates specified. See the **CEG221 Section 1 Home Page** for detailed requirements and instructions for turning in programming projects. No make-up exams will be given unless a serious illness or a bona fide emergency can be verified. Exceptions to the policy for turning in work late and giving make-up exams may be made in unusual circumstances when the student provides documentation in writing from an appropriate source. **All work must be your own; copying or sharing program code will constitute a breach of academic integrity and could result in failure of the course for all individuals involved.** Sharing

programming ideas and general computer skills with others outside of class is encouraged, especially through the Course Home Page on WINGS at <http://luminis1.wright.edu/cp/home/loginf>. See the handout for accessing the CEG221 Course Home Page on WINGS.

**CEG221 Section 1 Home Page:** <http://www.cs.wright.edu/~rhelt/ceg221/ceg221.html>

<b>Week</b>	<b>Topics - Readings in Schildt Text - Project Dues Dates - Exams</b>
1	Review of C Including Control Statements, Data Files, and User-defined Functions - Chapters 1-3, 6, 8-16
2	Array and Pointers - Chapters 4-5, 17
3	Structures, Unions, Enumerations, and typedef Statement - Chapter 7 - <b>Project No. 1 due on 18 Sep</b>
4	Sorting Algorithms, Part 1 - Chapter 21 - <b>Project No. 2 due on 25 Sep - Exam No. 1 on 28 Sep</b>
5	Sorting Algorithms, Part 2 and Complex Numbers - Chapter 21
6	Linked Lists - Chapter 22 - <b>Project No. 3 due on 9 Oct</b>
7	Double and Circular Linked Lists - Chapter 22 - <b>Project No. 4 due on 16 Oct - Exam No. 2 on 19 Oct</b>
8	Stacks and Queues - Chapter 22 and Hash Tables – Chapter 23
9	Hash tables - Chapter 23 and Binary Trees – Chapter 22, 23 – <b>Project No. 5 due on 30 Oct</b>
10	Binary Trees - Chapter 22, 23 and C++ Introduction – <b>Project No. 6 due 6 Nov</b>
	<b>Final Exam - Thursday, 16 Nov, 8:00 p.m. – 10:00 p.m. in Room RC152A</b>

**Schedule:** Topics and lecture dates may vary. Programming project due dates and exam dates are firm.